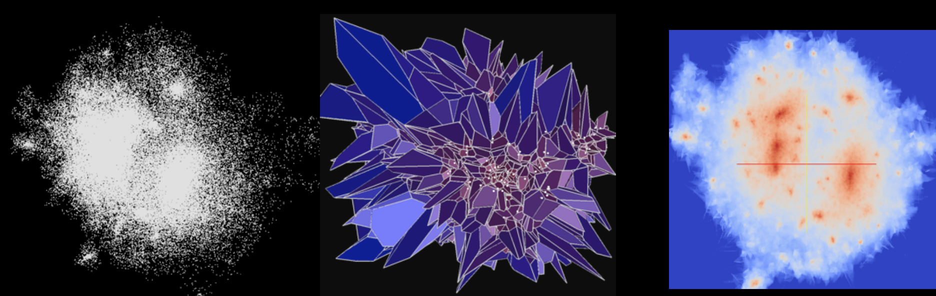


Adaptive Density Estimation of Particle Data

Tom Peterka[1], Steve Rangel[2], Nan Li[3],
Salman Habib[1], Katrin Heitmann[1]

[1]Argonne National Laboratory
[2]Northwestern University
[3]University of Chicago

Halo particles, Voronoi tessellation, and 2D density estimation



Tom Peterka
tpeterka@mcs.anl.gov
Mathematics and Computer Science Division

LANL Invited Talk
3/4/14

Executive Summary

We describe work in progress for sampling a regular density field from a distribution of particle positions using a Voronoi tessellation as an intermediate data model.

Key Ideas

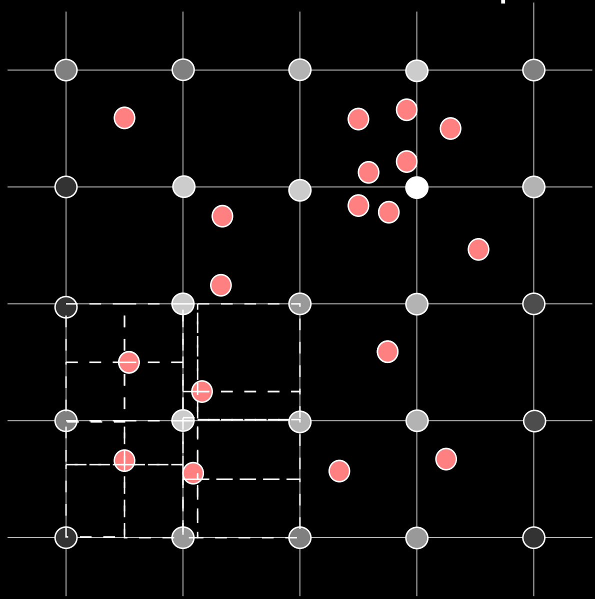
- Convert discrete particle data into continuous function that can be interpolated, differentiated, interpolated, represented as a regular grid (field)
- Automatically adaptive window size and shape
- Comparison with CIC and SPH using synthetic and actual data
- Voronoi tessellation and density estimation computed in parallel on distributed-memory HPC machines
- Application to gravitational lensing

Preliminaries

Estimation Kernels

CIC

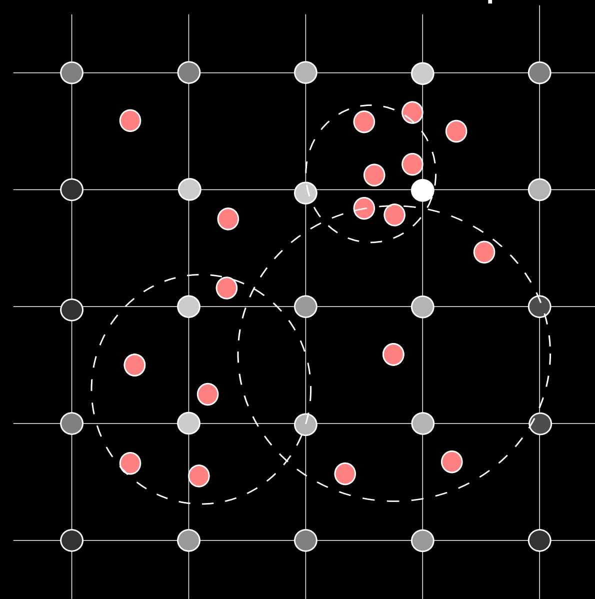
Fixed size and shape



In cloud-in-cell (CIC) methods, particles are distributed to a fixed number of grid points.

SPH

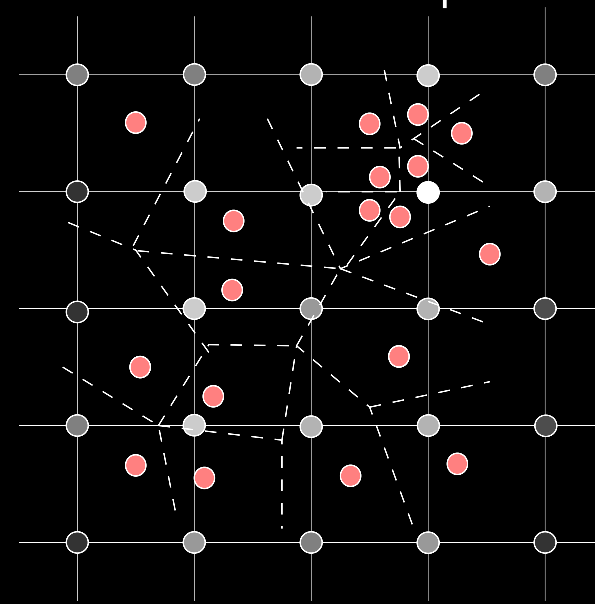
Variable size and fixed shape



In smoothed particle hydrodynamics (SPH) methods, particles are distributed to a variable number of grid points according to a variable size and fixed shape smoothing kernel.

TESS

Variable size and shape



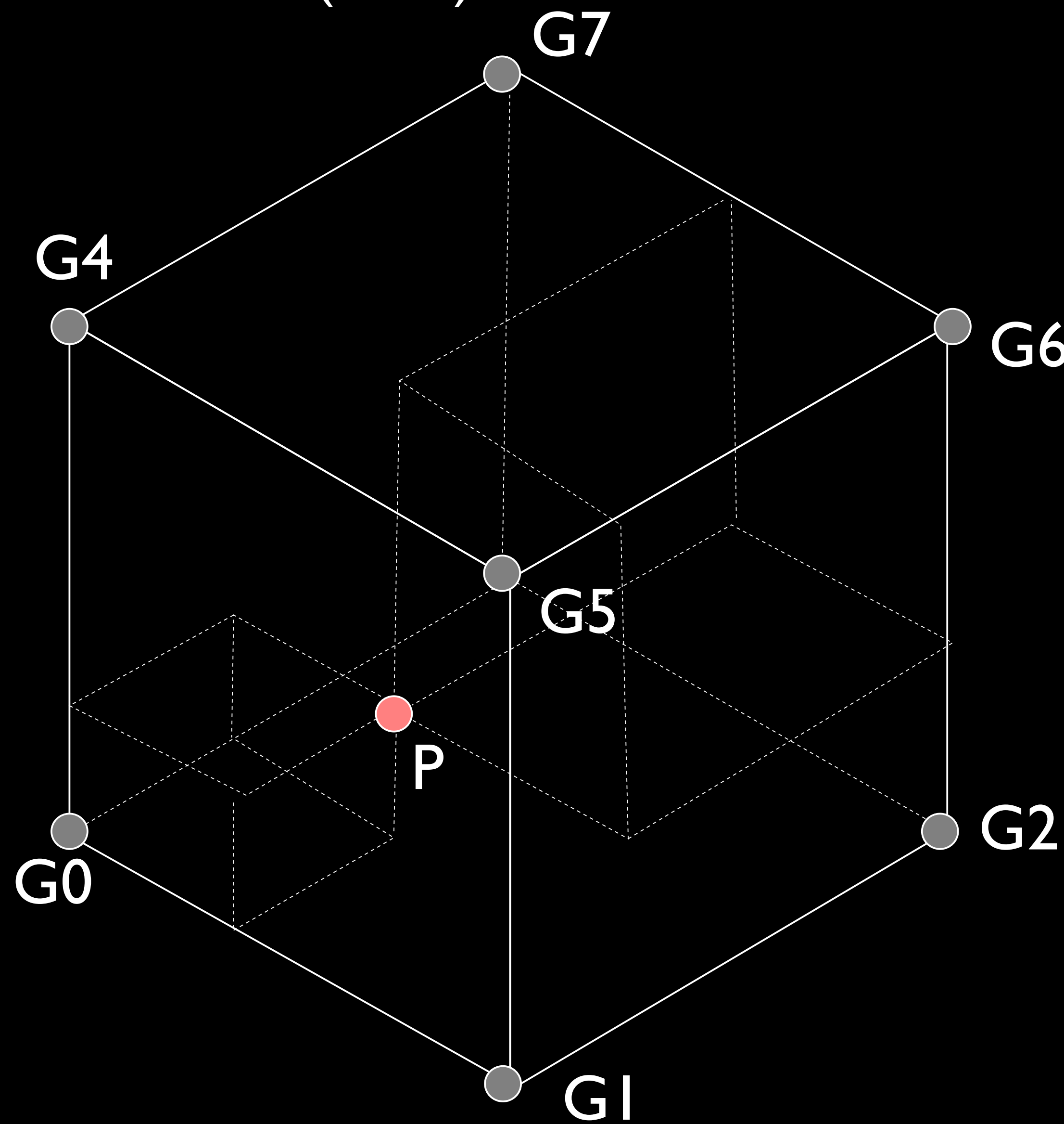
In tessellation (TESS) methods, particles are distributed to a variable number of grid points according to the Voronoi or Delaunay tessellation that has variable size and shape cells.

Cloud in Cell (CIC)

The mass of point P is distributed among nearest grid points $G_0 – G_7$.

The volume of of the grid cube with corners $G_0 – G_7$, $v(G_0 , G_7)$ is normalized to 1.0

The mass assigned to grid point G_i is $m(G_i) = 1.0 – v(G_i , P)$



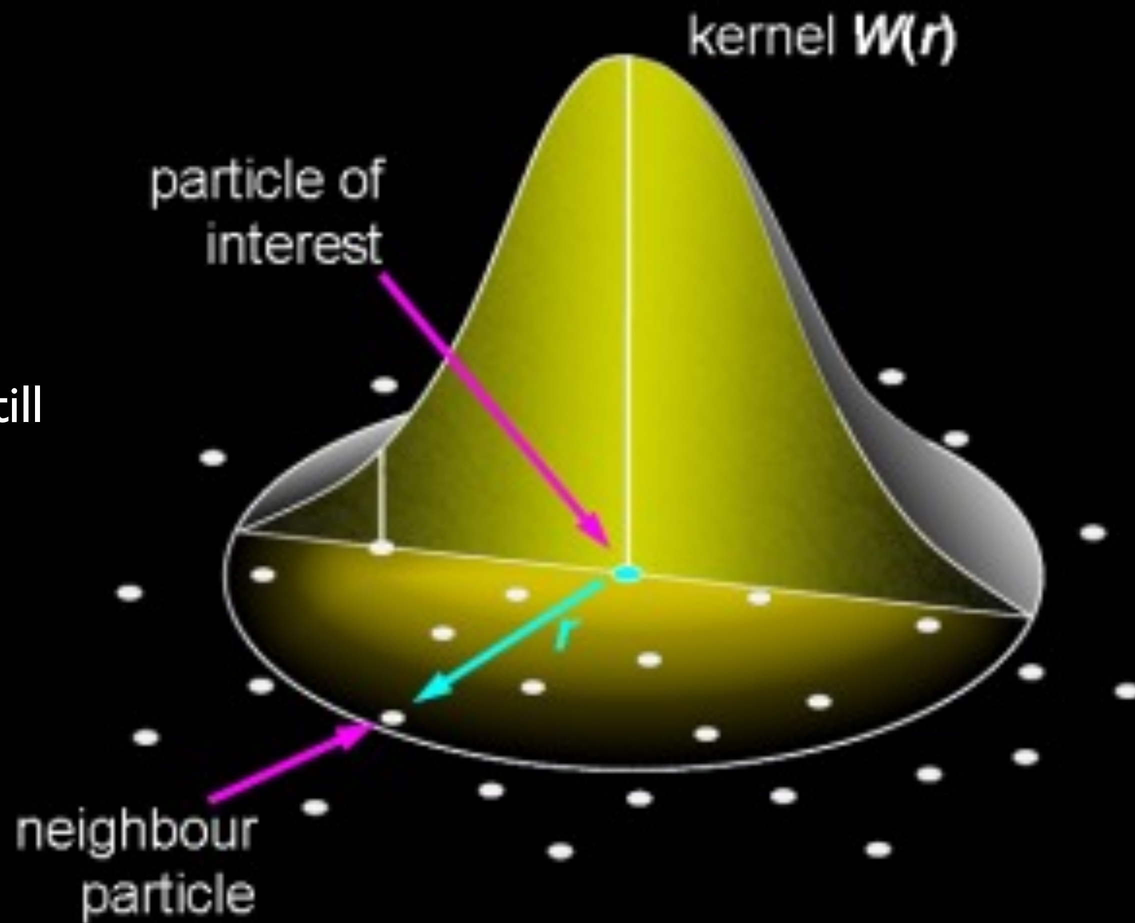
Smoothed Particle Hydrodynamics (SPH)

Size of kernel is determined by particle density, not by grid spacing (eg. radius of *n* particles)

n is a parameter that must still be determined a priori

Kernel *W(r)* also must be specified, eg. Gaussian

Shape is symmetrical, eg. spherical



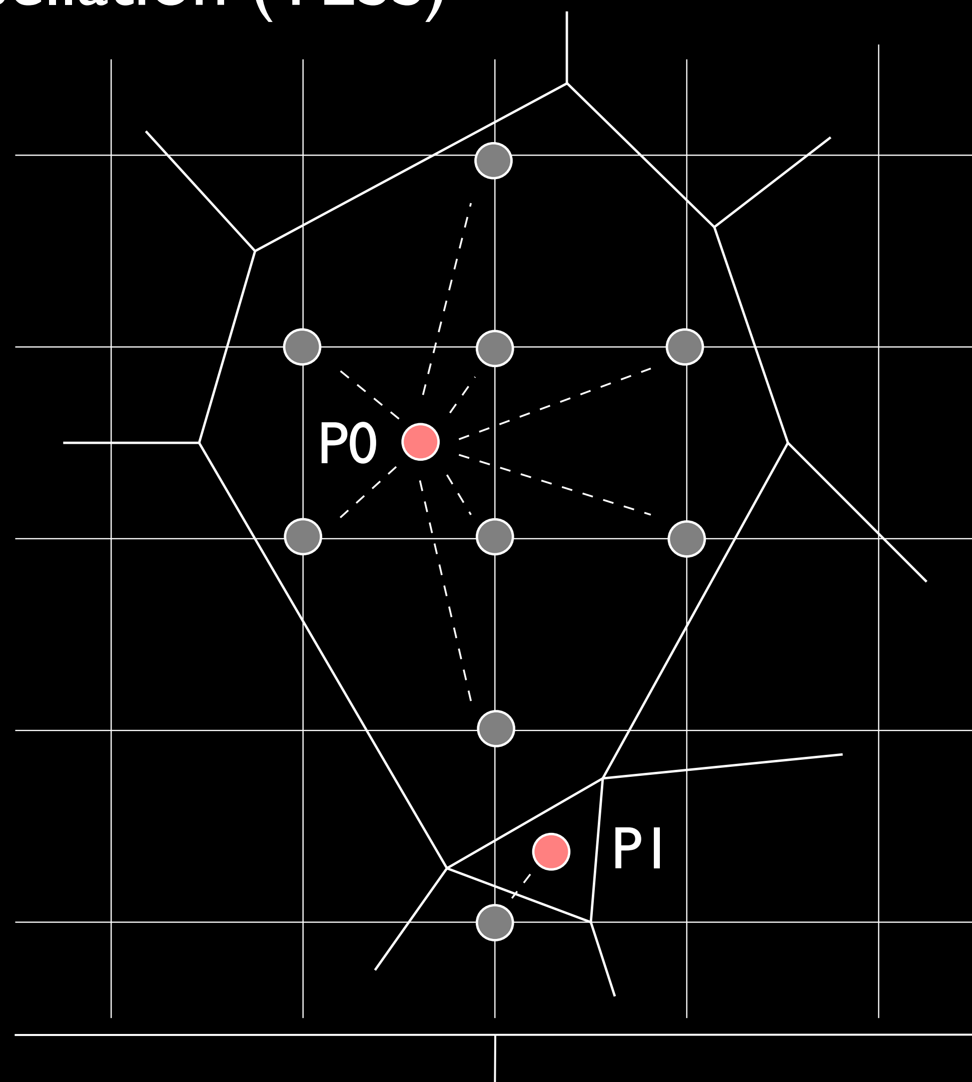
Tessellation (TESS)

Parameter free: no fixed window size determined by grid or number of particles

Kernel free: no smoothing kernel

Shape free: asymmetrical, no window or kernel shape

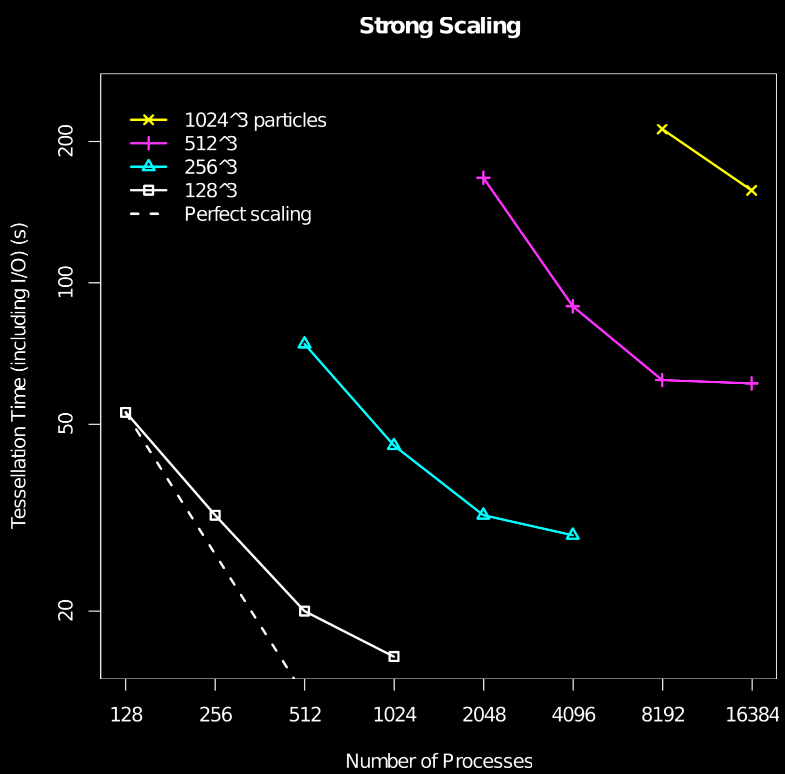
Automatically adaptive



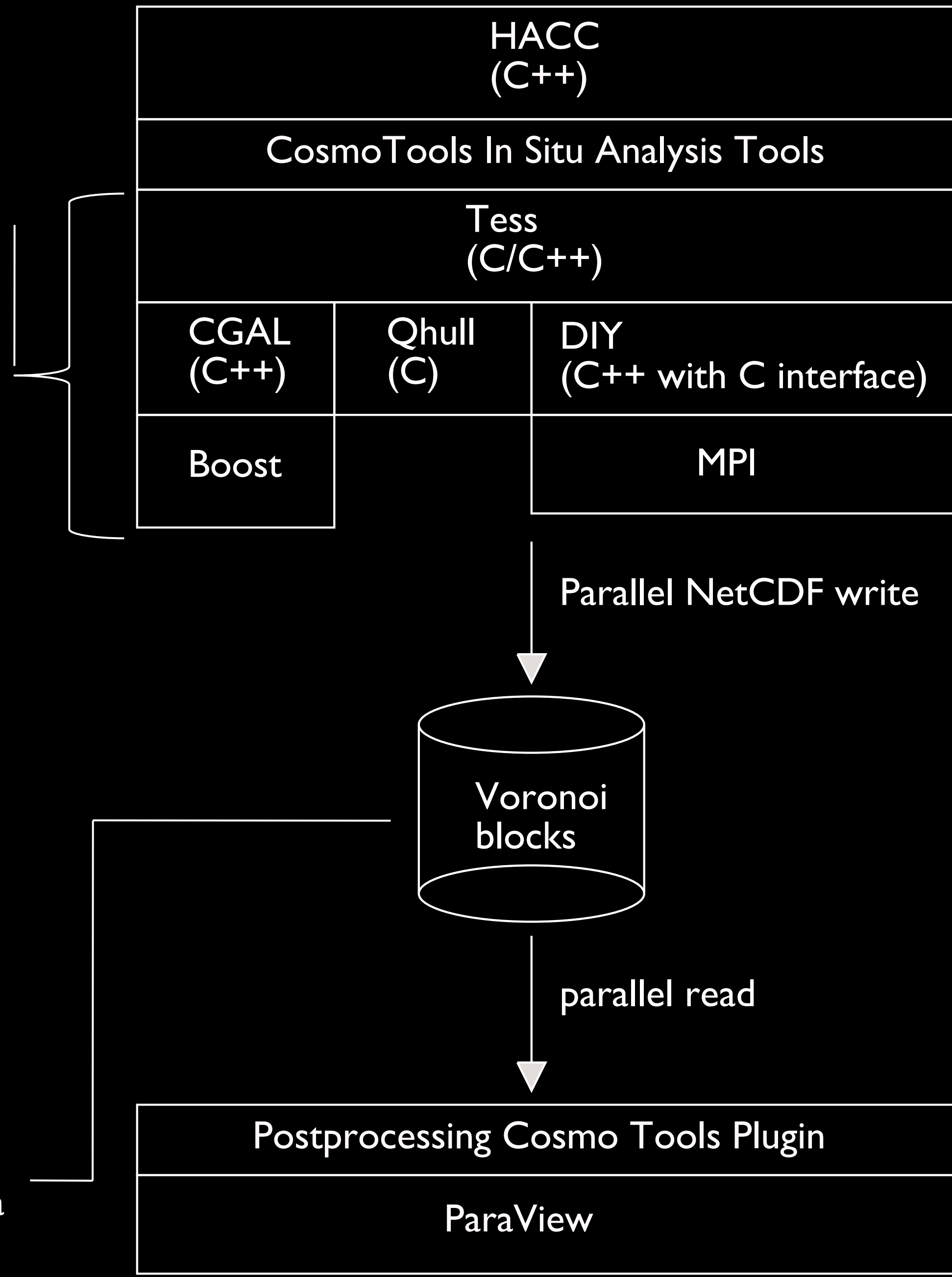
P0 is a particle whose Voronoi cell covers several grid points. Its mass is uniformly distributed (zero-order estimation) to those grid points. P1 is a small cell that covers no grid points. Its mass is assigned to the nearest grid point.

Tess Library

Tess is our parallel library for large-scale distributed-memory Voronoi and Delaunay tessellation.



Dense, our density estimator, currently reads the tessellation from disk and estimates density onto a regular grid. Eventually dense will be converted to a library that can be coupled in memory to tess output, saving the tessellation storage.



DIY Library

Features

-Parallel I/O to/from storage

-Domain decomposition

-Network communication

-Written in C++, with C-style bindings, can be called from Fortran, C, C++

-Autoconf build system

-Lightweight: libdiy.a 800KB

-Maintainable: ~15K LOC

-MPI + openmp hybrid parallel model

Benefits

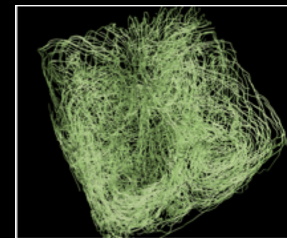
-Enable large-scale data-parallel analysis on all HPC machines

-Provide internode scalable data movement

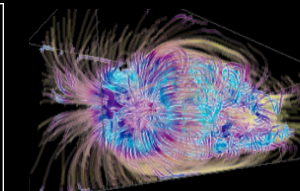
-Analysis applications can be custom

-Reuse core components

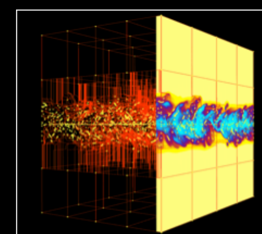
Applications



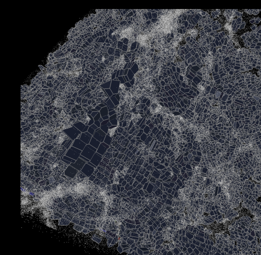
Particle tracing in thermal hydraulics



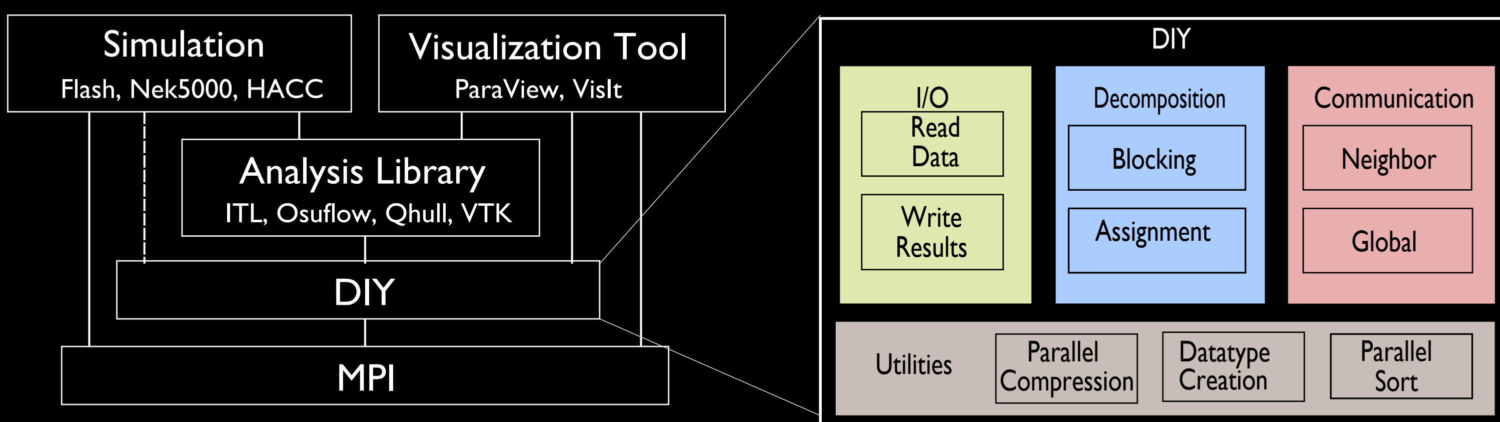
Information entropy in astrophysics



Topology in combustion



Computational geometry in cosmology

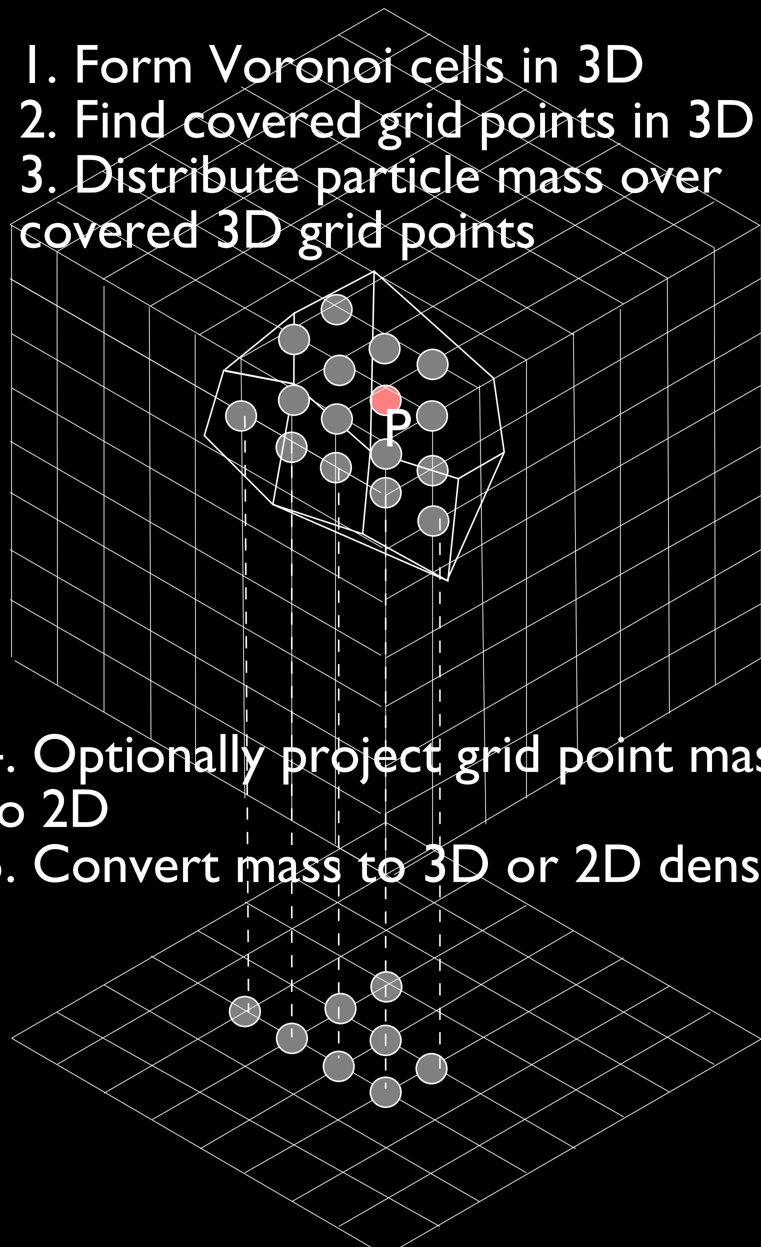


DIY usage and library organization

Method

Overall Algorithm

1. Form Voronoi cells in 3D
2. Find covered grid points in 3D
3. Distribute particle mass over covered 3D grid points



4. Optionally project grid point mass to 2D
5. Convert mass to 3D or 2D density

```
for (all Voronoi cells) {

  compute grid points in cell bounding box

  compute Voronoi cell interior grid points from
     grid points in cell bounding box

  for (all interior grid points) {

    if (grid point is in bounds of local block)
      add mass contribution to grid point

    else
      send mass contribution to neighboring block
         containing grid point and add it there

    if (no grid points in interior of Voronoi cell)
      add mass contribution to single nearest
         grid point

    if (2D projection) {
      accumulate mass at 2D pixel
      divide by pixel area for 2D density
    }

    else
       divide by voxel volume for 3D density

  } // interior grid points

} // Voronoi cells
```

Complexity and Optimizations

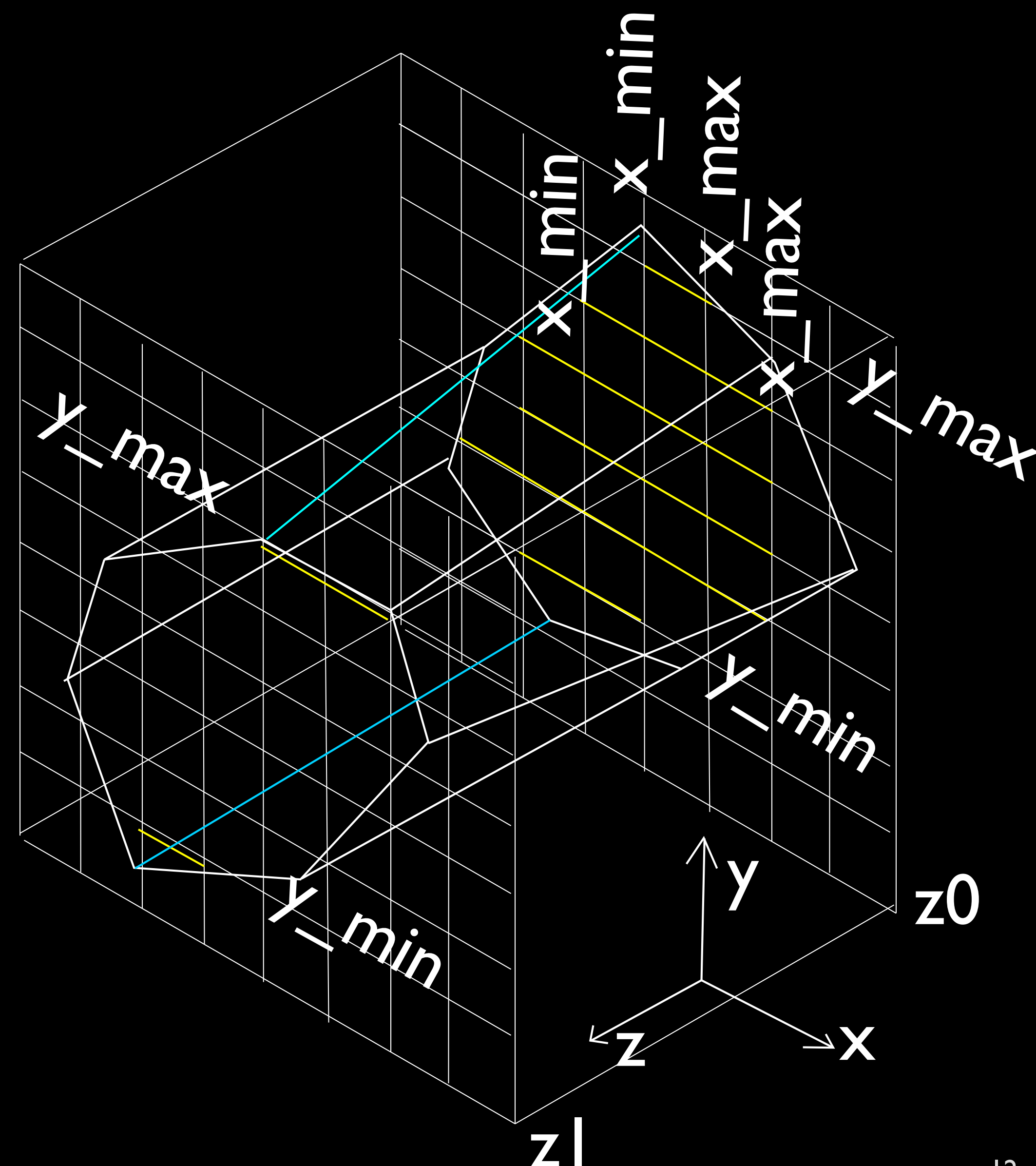
Naïve algorithm to find interior grid points of each Voronoi cell (polyhedron) is O(n^3). Triple nested loop

```
for all z, {
   for all y {
     for all x {
        scan line search for border
```

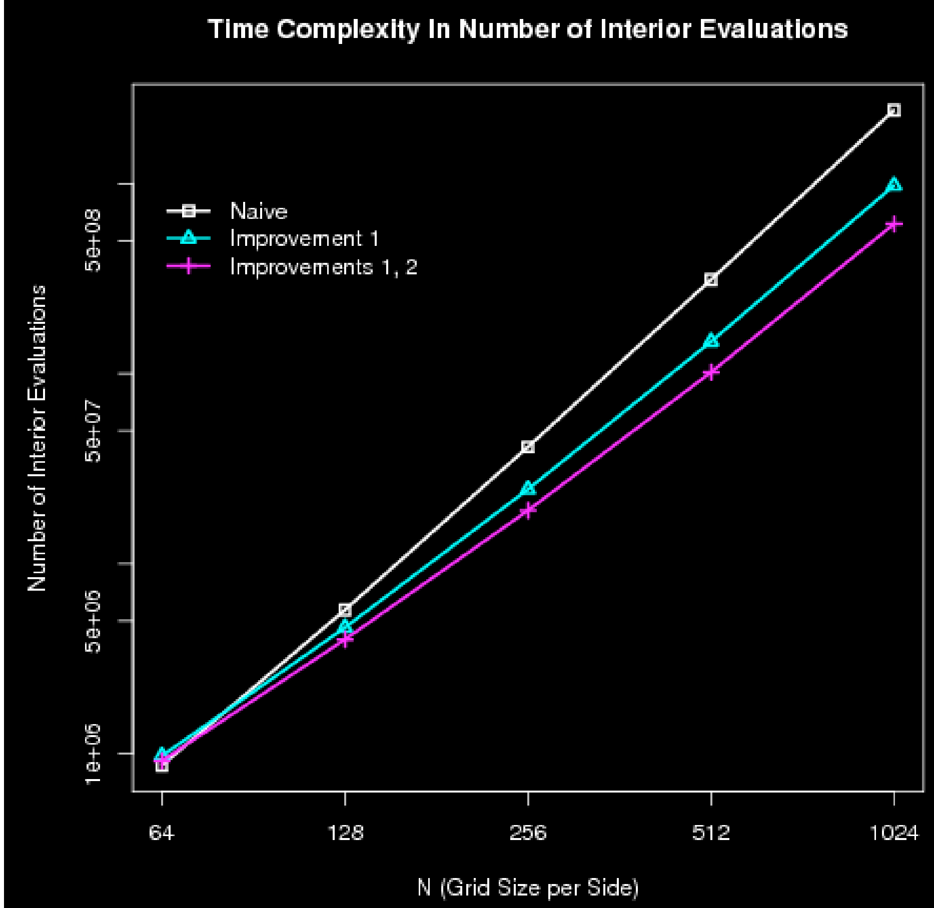
n is size of grid in one dimension since bounding boxes tessellate the grid (plus some overlap in cell bounds)

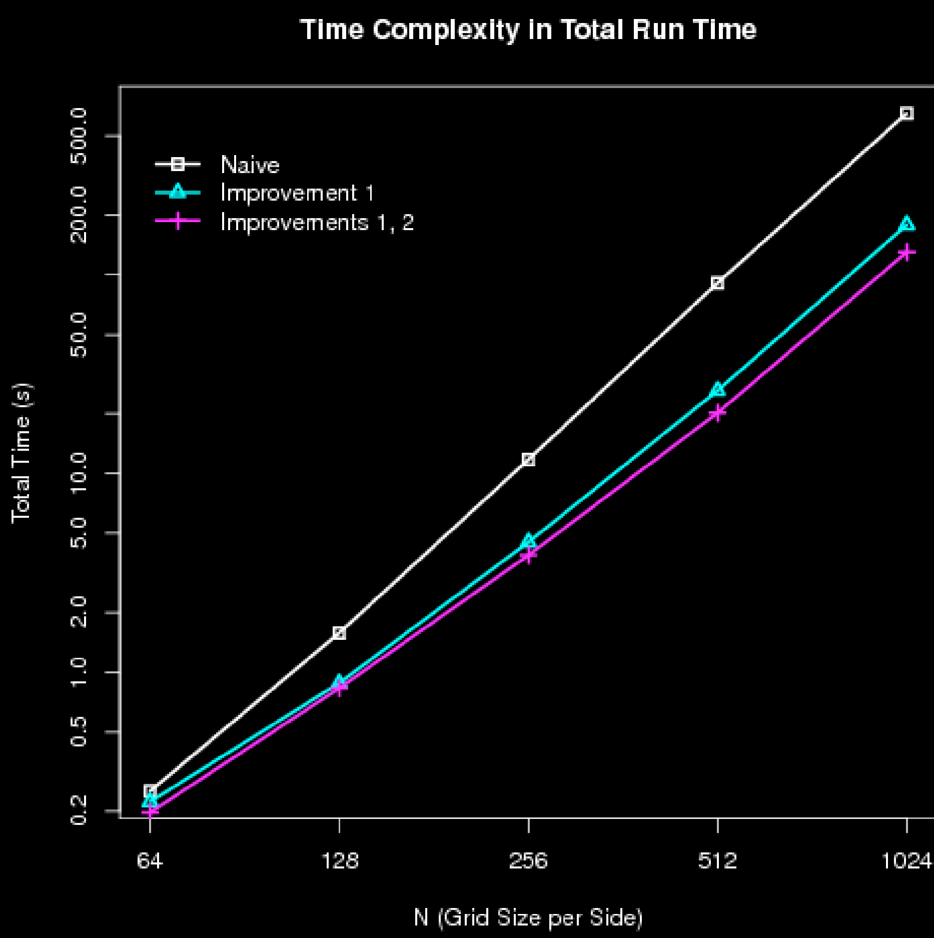
Limit grid point search:

- Limit x scans: don't need to find interior, only cell boundary crossings, and can use previous scan boundary crossings as starting points for next scan
- Limit y scans: use y limits at previous z as starting y coordinates of next set of scans



Optimizations



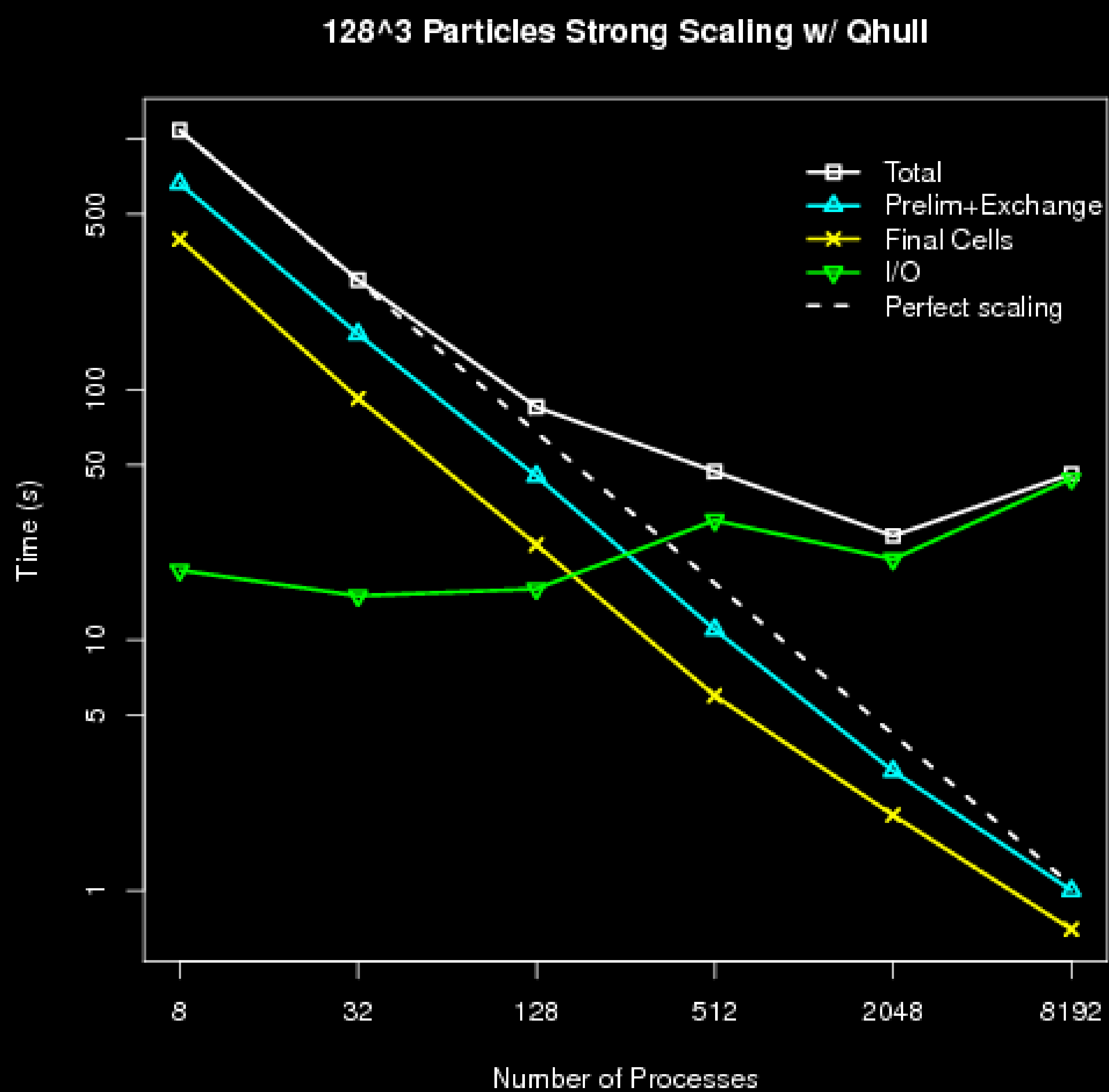


Time complexity as a function of number of interior evaluations for different grid sizes

Run time for naïve and improved algorithms is bounded by number of interior evaluations

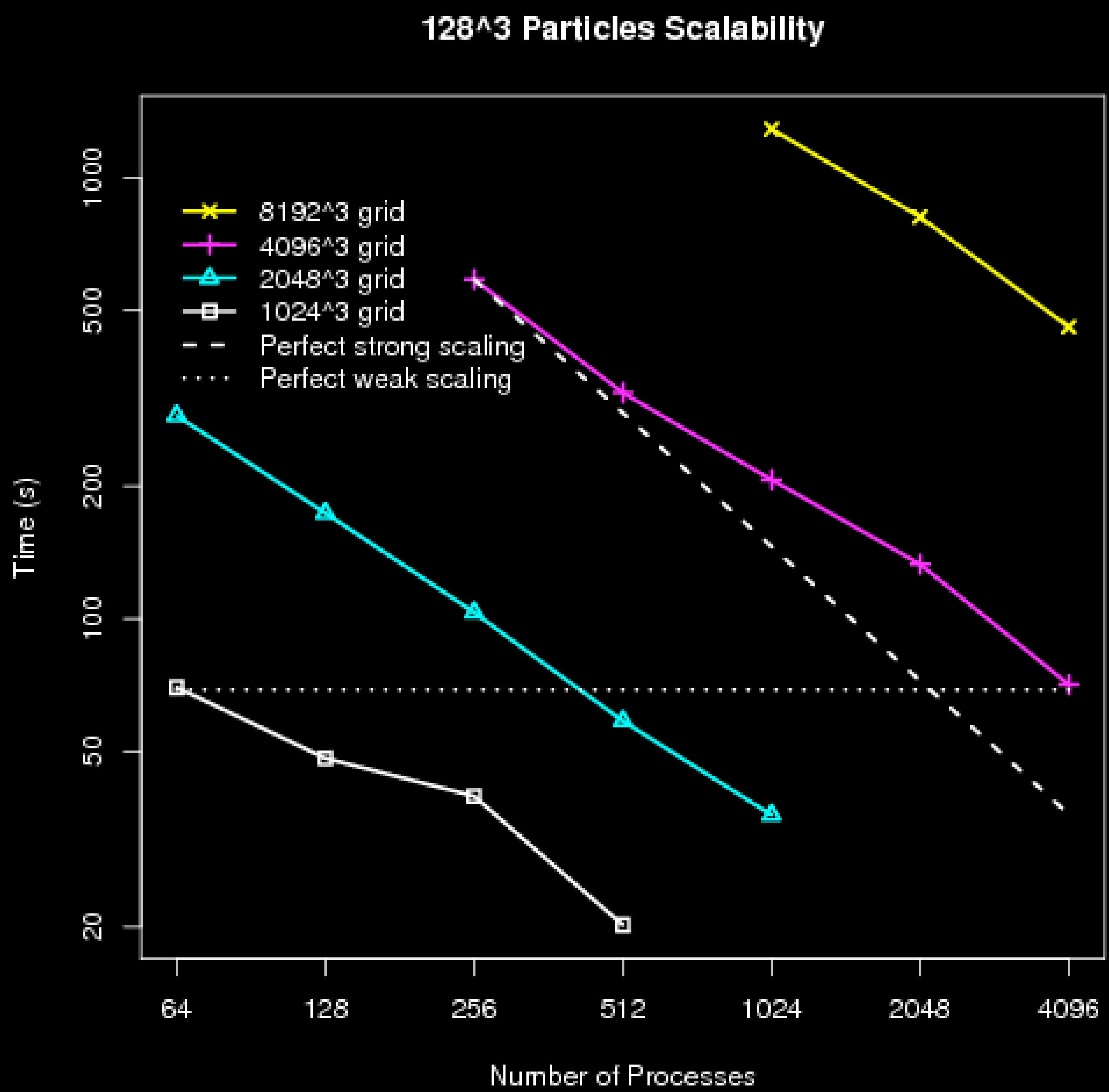
Tess Strong Scaling

- 128^3 synthetic particles
- End-to-end time and component times shown
- 60% strong scaling excluding I/O



Dense Strong and Weak Scaling

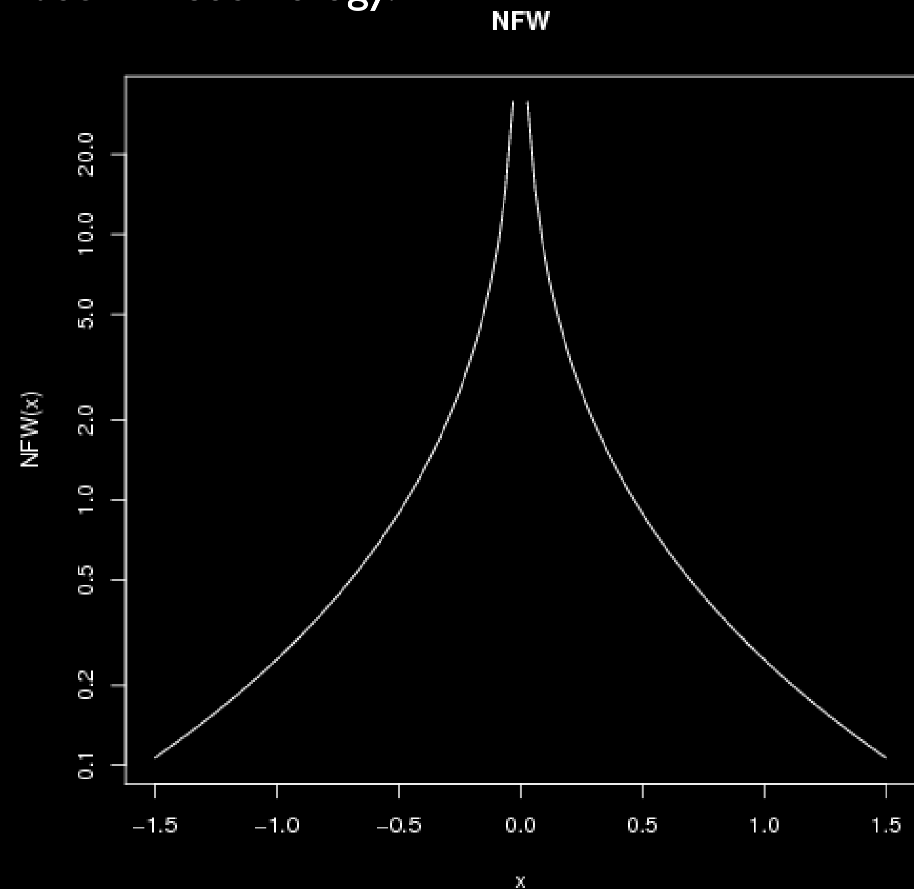
- 128^3 synthetic particles
- End-to-end time (including reading tessellation and writing image)
- 3D->2D projection
- 51% strong scaling (End-to-end) for 4096^3 grid



Accuracy

Navarro-Frenk-White (NFW)

Our first synthetic dataset is derived from an analytical density function commonly used in cosmology.



k is a constant, 1 for us

ρ (*r*) is Monte Carlo sampled to get test set of particles

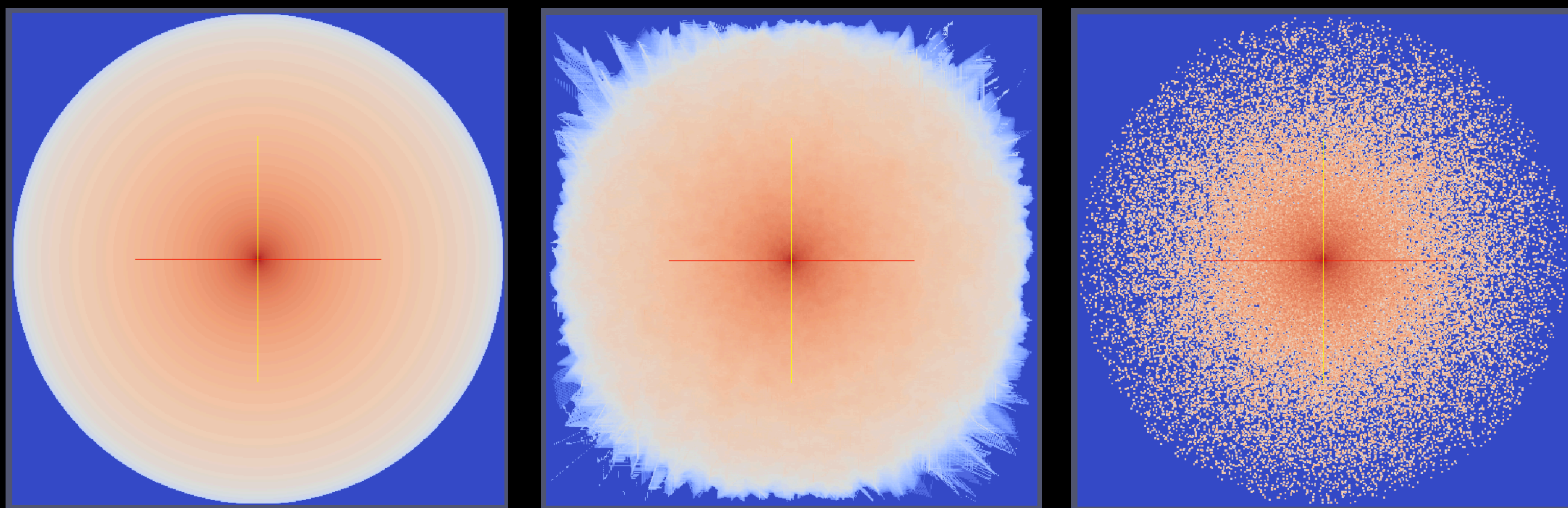
Ground truth is 2D plot of ρ (*r*)

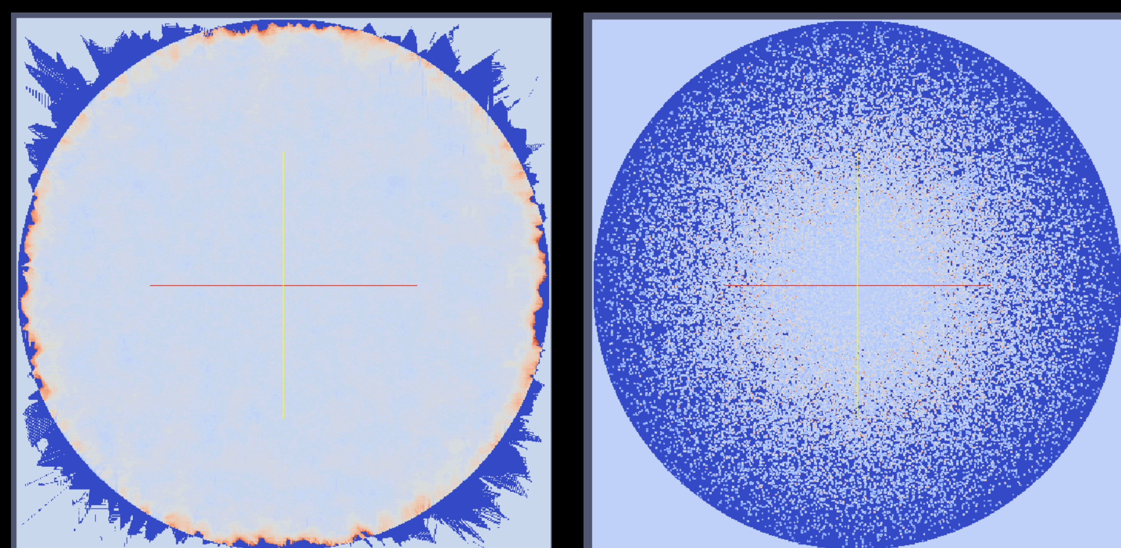
We limit *r* to [-1.5, 1.5] and NFW(*r*) to 10^6

$$\rho(r) = \frac{k}{(r(r+1)^2)}$$

NFW 2D Density Fields

Analytical TESS CIC



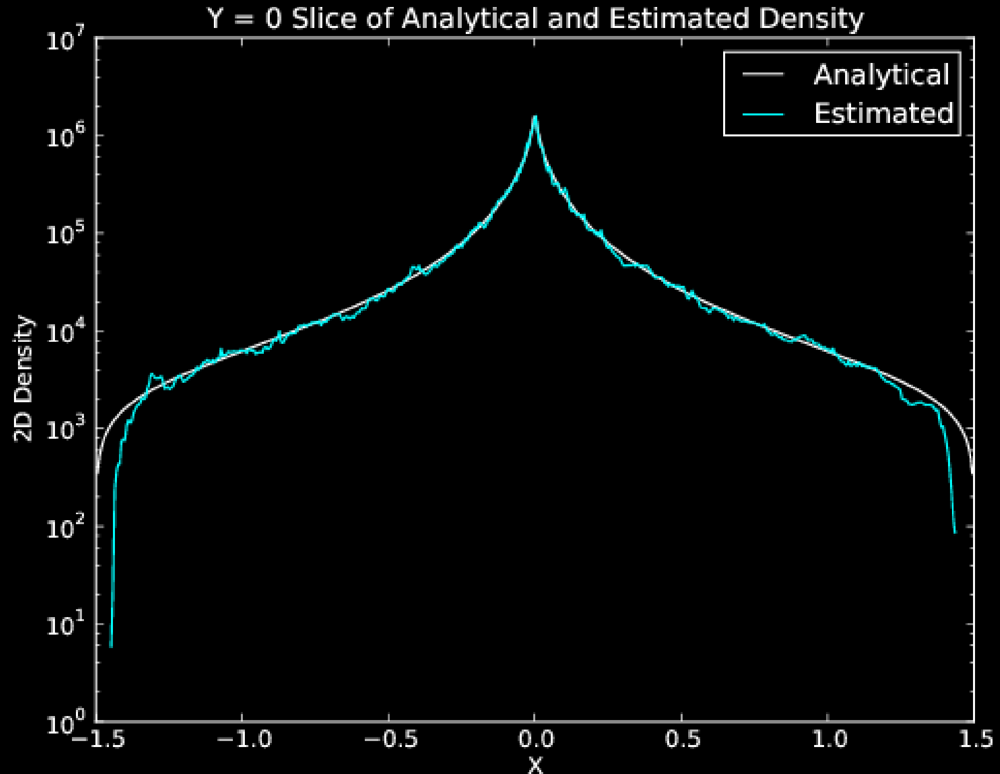


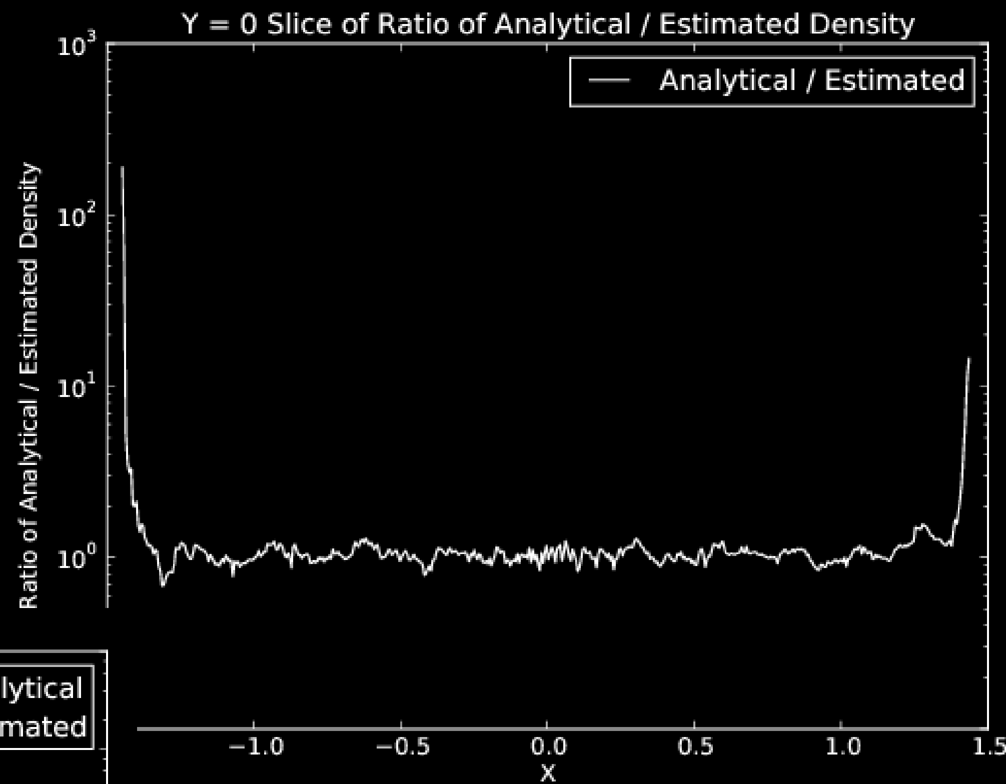
Top row:
1024^3 3D density projected to 1024^2 2D density field and rendered in ParaView

Bottom row:
Ratio of analytical divided by estimated density

TESS

Comparison between analytical 2D density and estimated density at $y = 0$ cross section

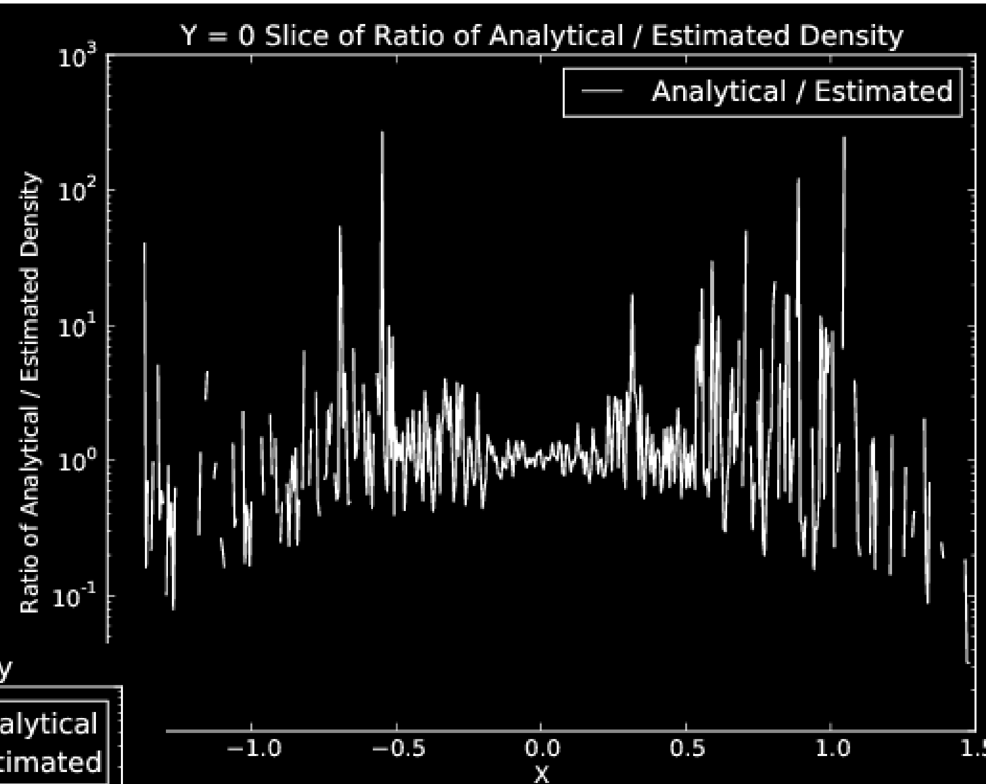


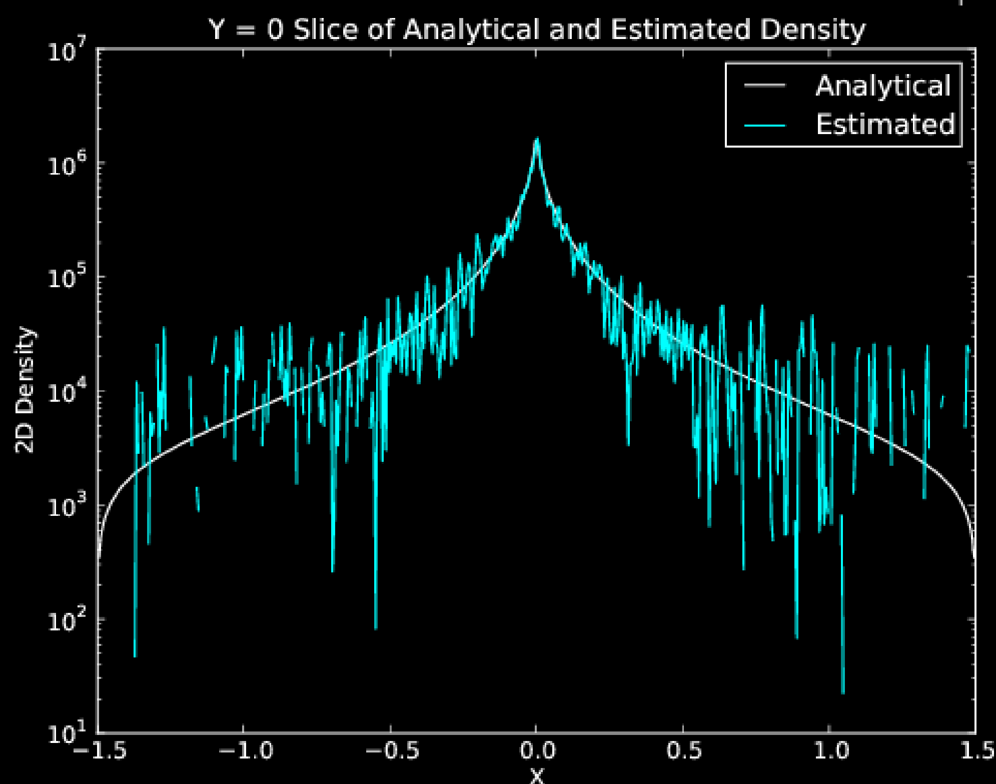


Ratio between analytical 2D density divided by estimated density at $y = 0$ cross section

CIC

Comparison between analytical 2D density and estimated density at $y = 0$ cross section

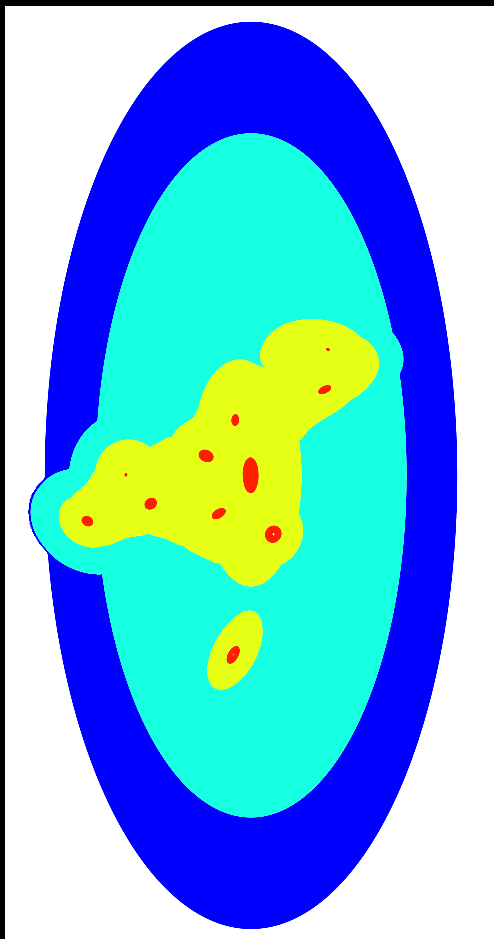




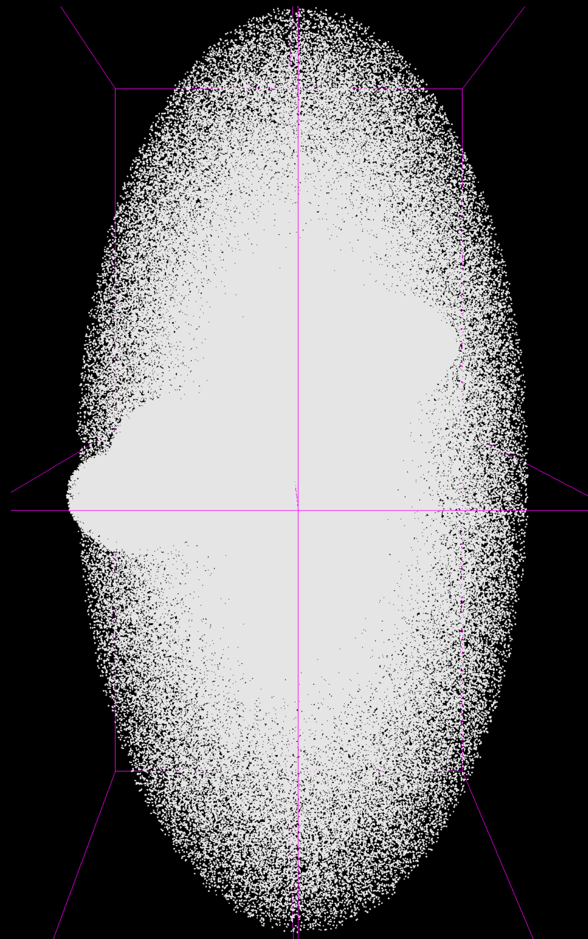
Ratio between analytical 2D density divided by estimated density at $y = 0$ cross section

Complex NFW (CNFW)

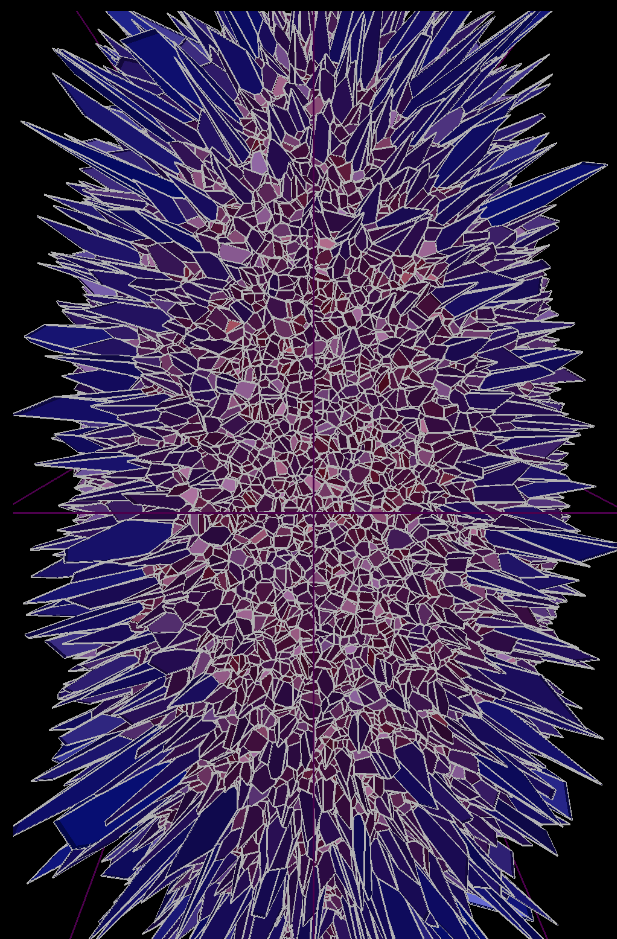
Our second synthetic dataset is a combination of several NFWs of varying cutoff densities and asymmetric scaling factors.



Analytical cutoff density contours

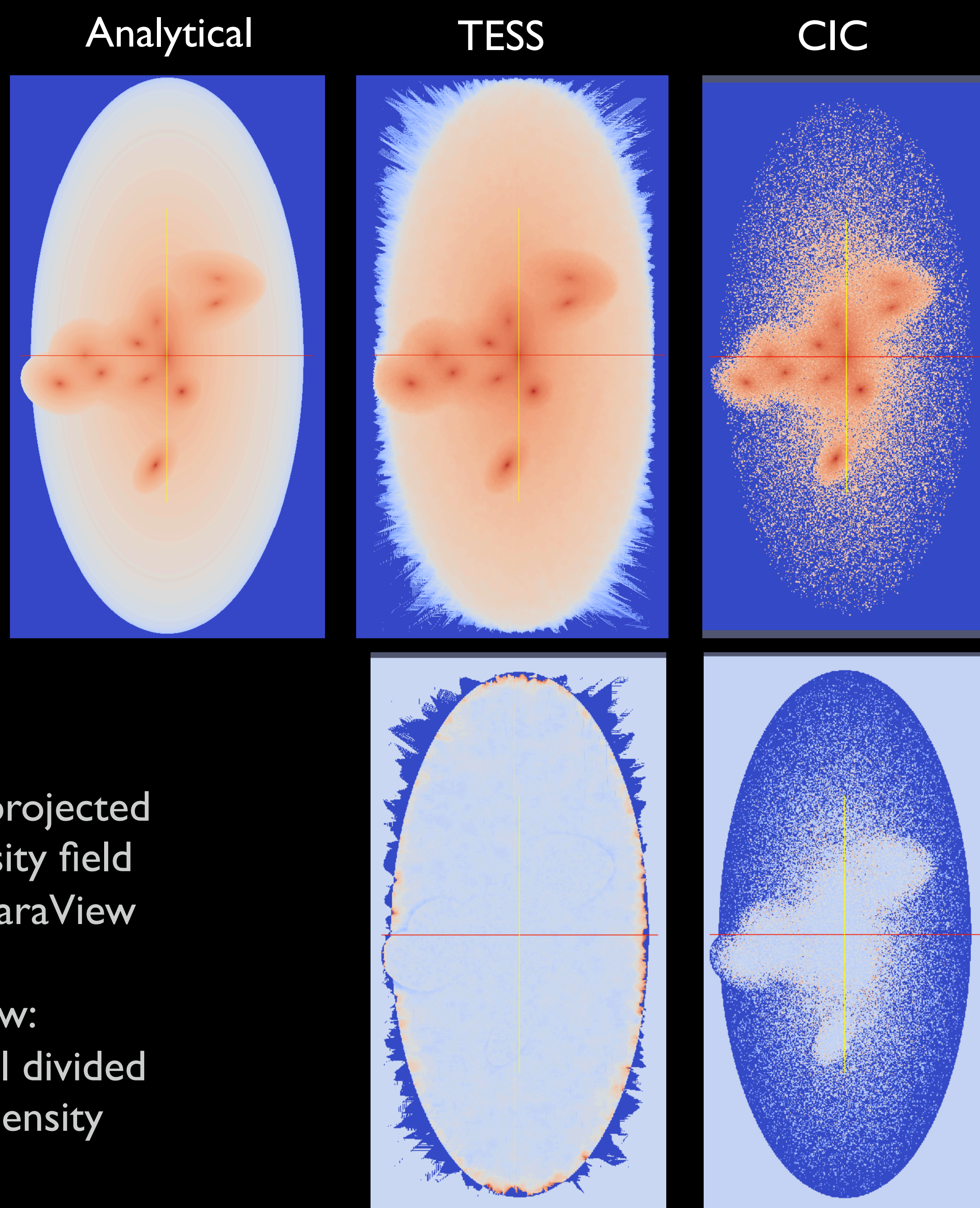


2e5 sampled particles



Voronoi tessellation

CNFW 2D Density Fields

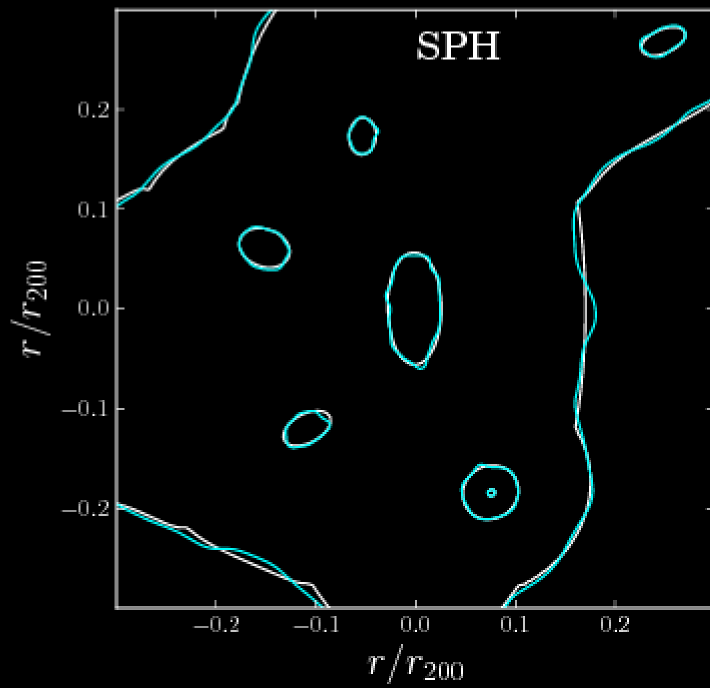


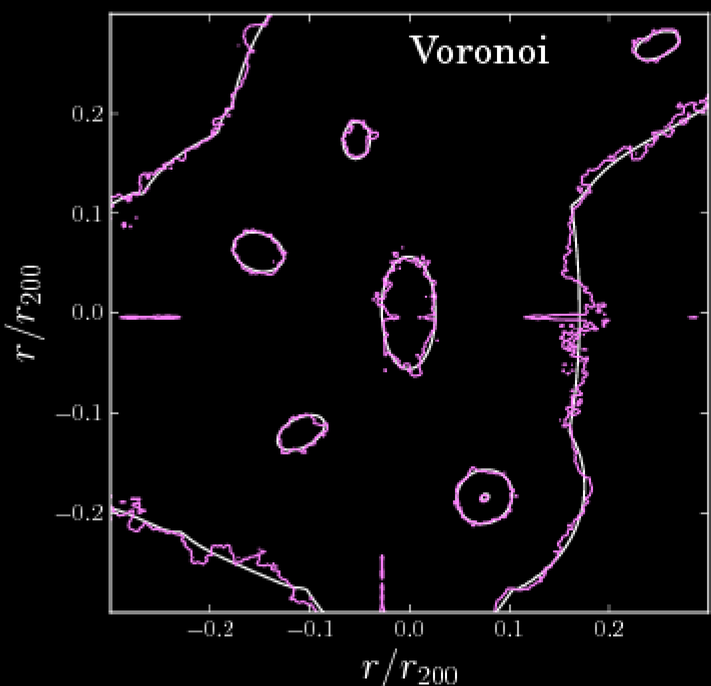
Top row:
1024^3 3D density projected
to 1024^2 2D density field
and rendered in ParaView

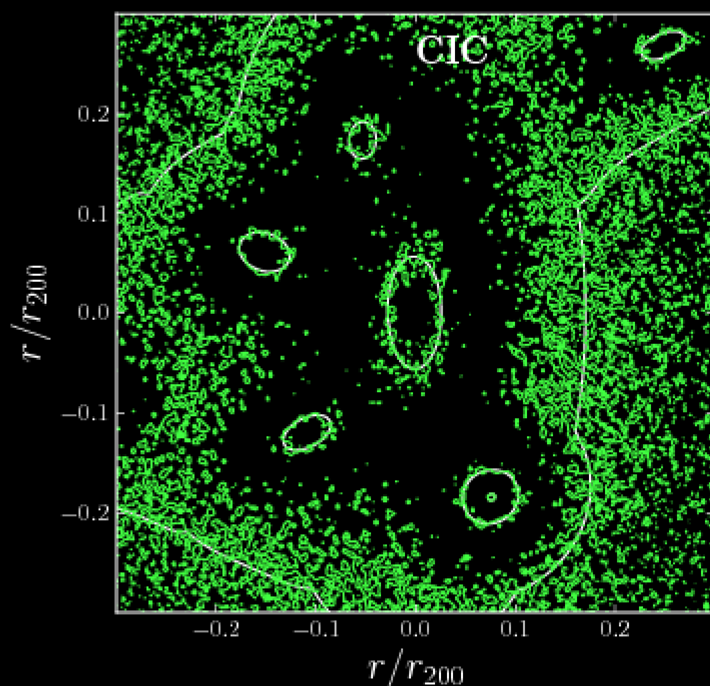
Bottom row:
Ratio of analytical divided
by estimated density

Isocontours

Isocontours taken at a target density value near the center of the CNFW dataset are another comparison of estimation methods. Upper right: SPH. Lower right: CIC. Lower left: TESS.

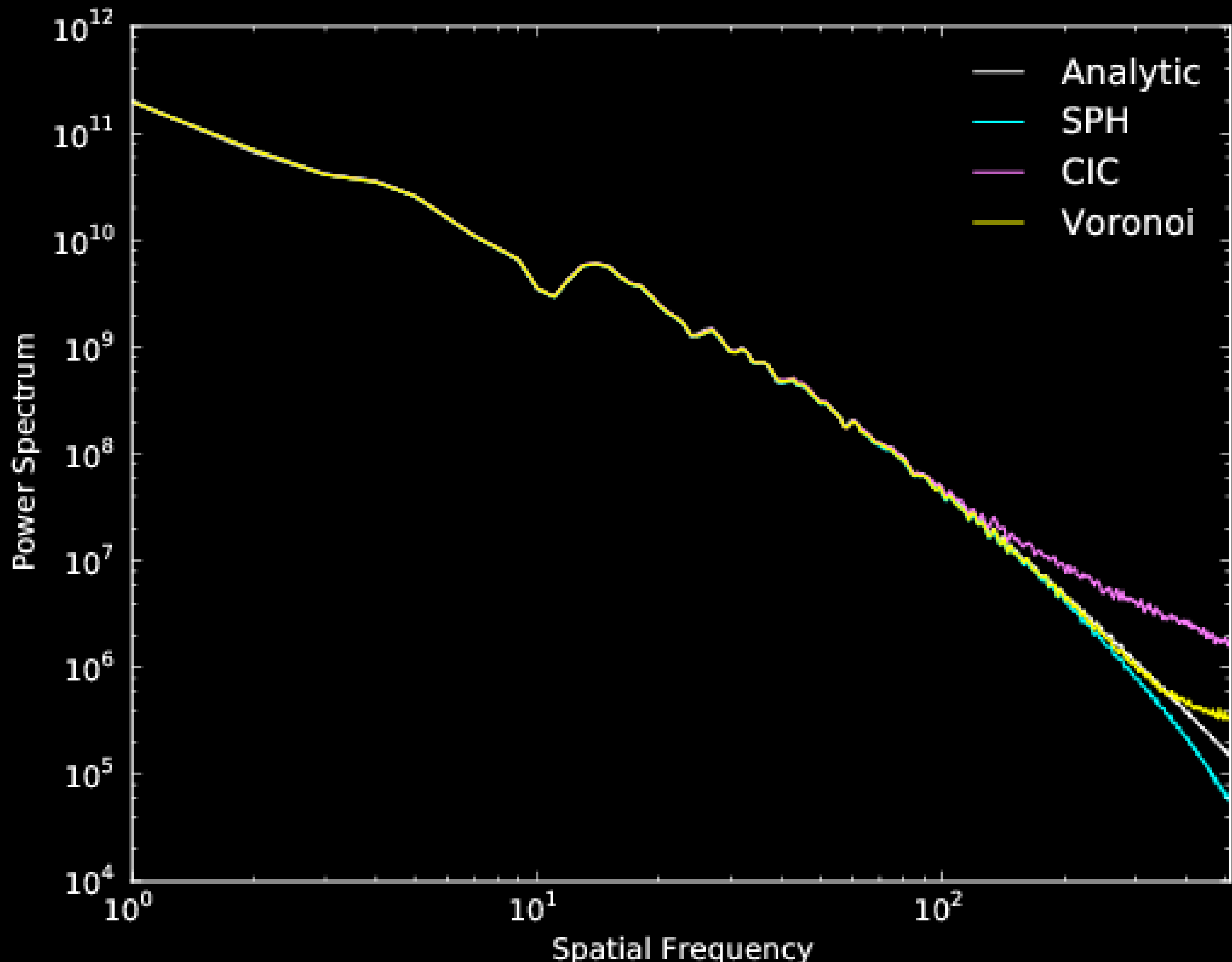






Density Power Spectrum

CNFW density power spectrum is derived from FFT of density and shows amount of density contained at different spatial frequencies. All methods do well at low frequencies, but diverge from analytical in high frequency regions.

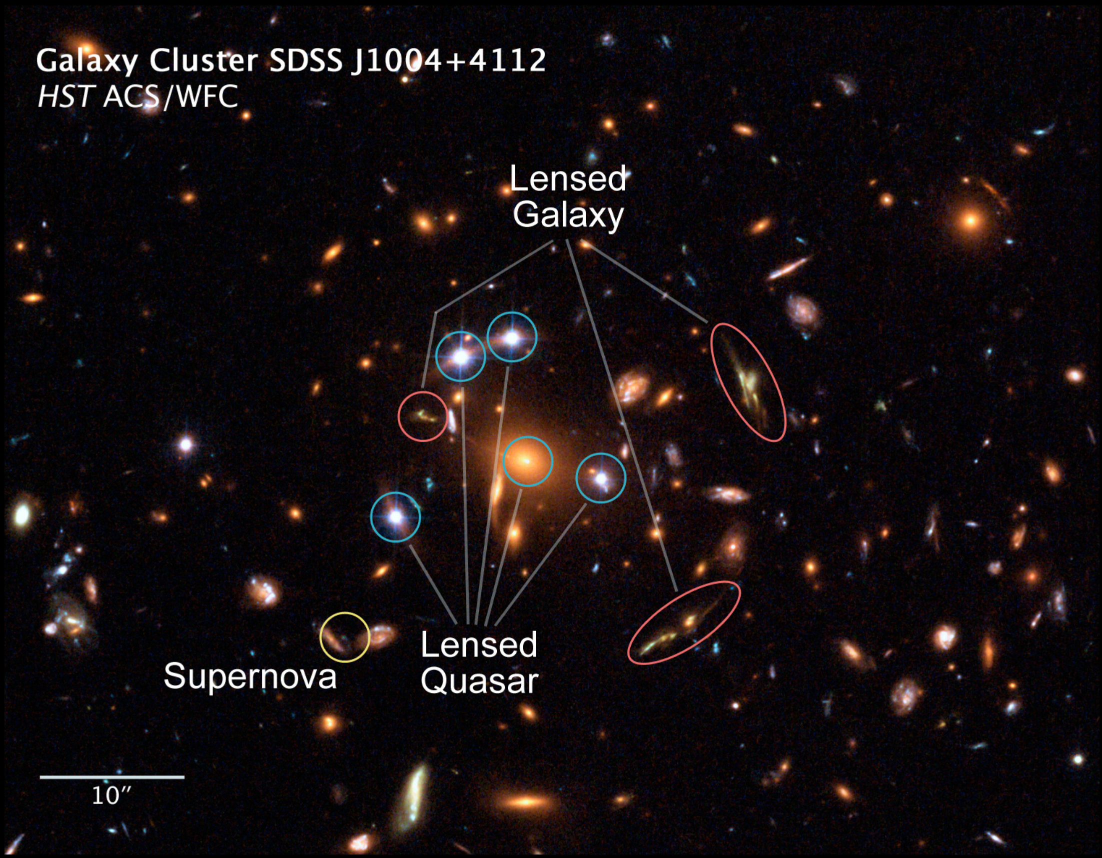


Application: Gravitational Lensing

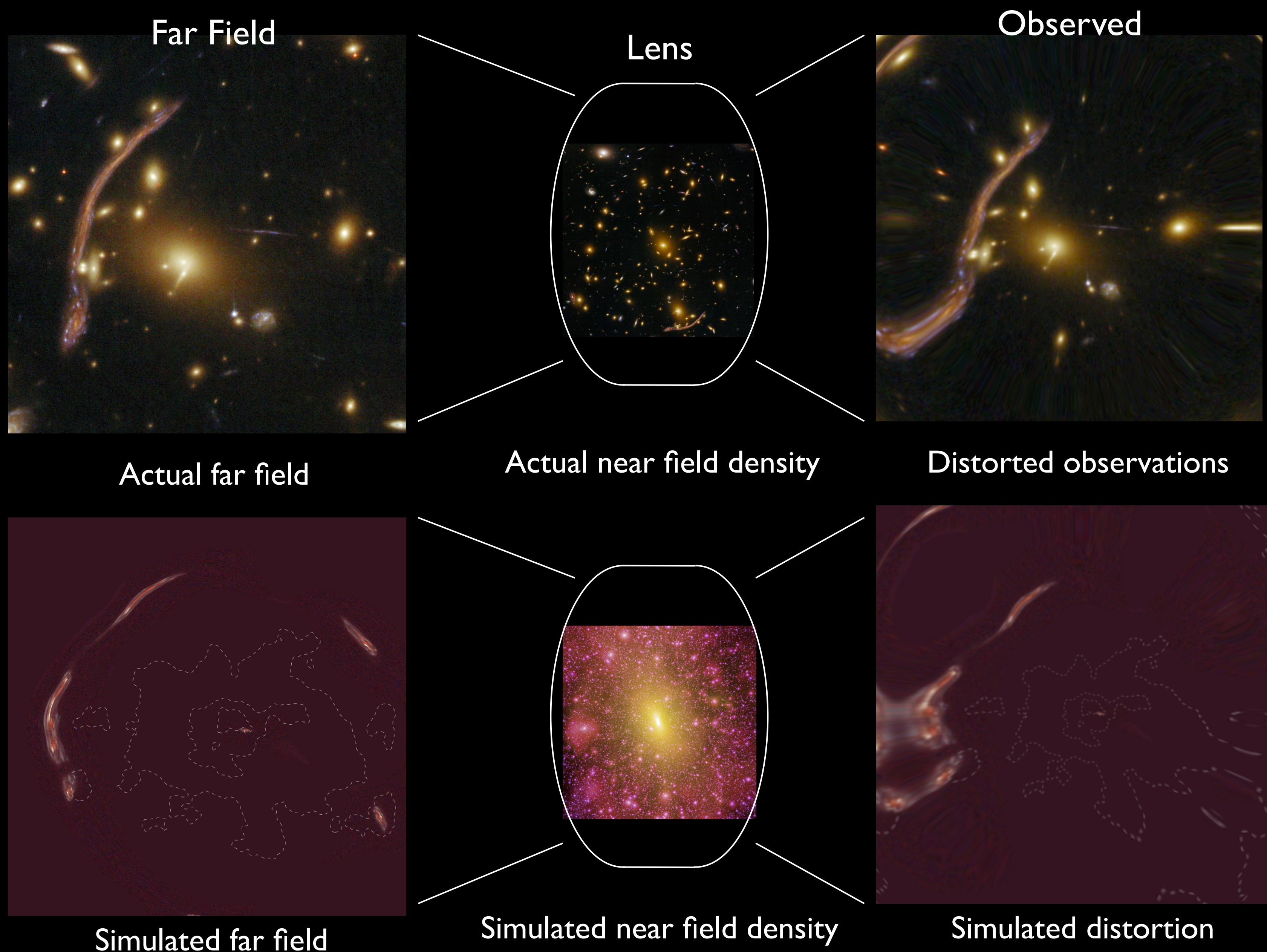
Lensing

One application of the density estimator is gravitational lensing for simulating the distortion of sky surveys as light rays are refracted by galaxies en route to the observer.

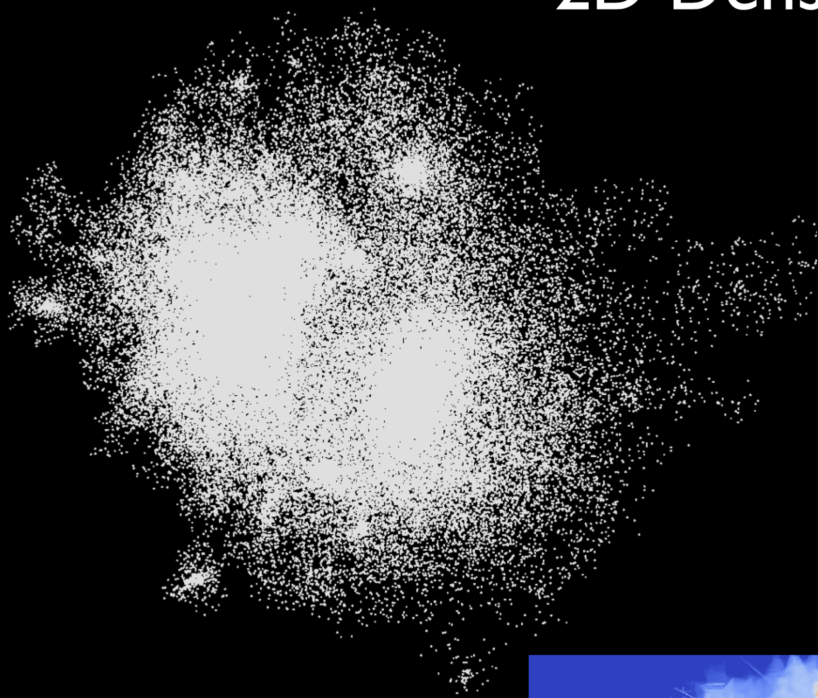
- Gravitational lensing = light rays deflecting when passing through a gravitational potential
- Properties of lensed images a a function of the gravitational potential between object and observer
- Can model gravitational potential as a 2D image of density of dark matter tracers

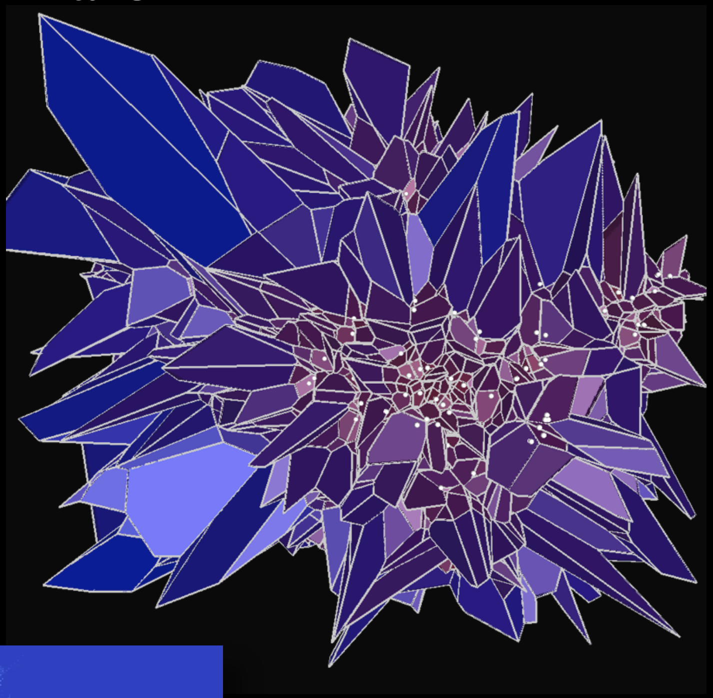


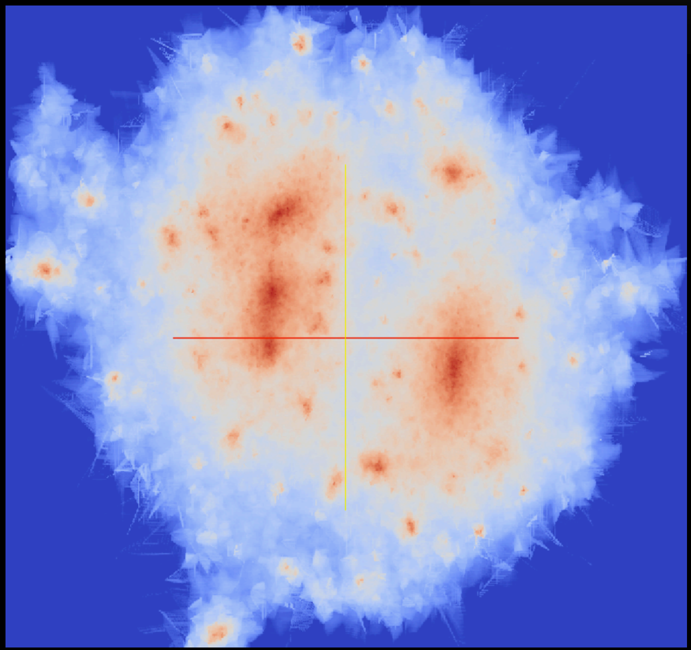
Lensing for Validating Simulations with Sky Surveys



2D Density of Halo







Particle data from HACC N-body cosmology code from halo ID 7445077095

Voronoi tesellation of halo particles colored by cell volume

Final output 2D density field for lensing

Summary

I described sampling a regular density field from a distribution of particle positions using a Voronoi tessellation as an intermediate data model.

Key Ideas

- Automatically adaptive window size and shape
- Comparison with CIC and SPH using synthetic and actual data
- Voronoi tessellation and density estimation computed in parallel on distributed-memory HPC machines
- Application to gravitational lensing

Ongoing and Future Work

- Linear Barycentric interpolation inside Voronoi cells through Delaunay tessellation
- Shared memory threading inside MPI tasks
- Other applications such as 3D volume rendering

"The purpose of computing is insight, not numbers."

–Richard Hamming, 1962

Acknowledgments:

Facilities

Argonne Leadership Computing Facility (ALCF)

Funding

US DOE SciDAC SDAV Institute

People

Juliana Kwan, Hal Finkel, Adrian Pope, Nick Frontiere, George Zagaris

Software

https://repo.anl-external.org/repos/tess/trunk

Tom Peterka

tpeterka@mcs.anl.gov

Mathematics and Computer Science Division